Functions & Recursion (1B)

f3.c

```
#include <stdio.h>
#include <string.h>

void reverse( const char * const p) {
  printf("given p= \"%s\"\n", p);
  if ('\0' == p[0]) {
    printf("reverse(\"\") returns \n");
    return;
  }
  else {
    printf("reverse(\"%s\") + %3c \n", &p[1], p[0]);
    reverse( &p[1]);
    putchar( p[0] );
  }
}


int main(void) {
  char s[] = "ABCDEFG";
  int  i;

  printf("sizeof(s)= %ld \n", sizeof(s));

  for (i=0; i<sizeof(s); ++i) {
    printf("s[%d]= 0x%02x %2d %c\n", i, s[i], s[i], s[i]);
  }

  reverse(s);
  printf("\n");

}
```

f4.c

f4.c

```
#include <stdio.h>

int add(int a, int b);
int sub(int a, int b);
int mul(int a, int b);
int div(int a, int b);


int main(void) {
  int (*fp) (int a, int b);

  fp = &add;
  printf(" (*fp)(33,11)= %d \n", (*fp)(33,11) );

  fp = &sub;
  printf(" (*fp)(33,11)= %d \n", (*fp)(33,11) );

  fp = &mul;
  printf(" (*fp)(33,11)= %d \n", (*fp)(33,11) );

  fp = &div;
  printf(" (*fp)(33,11)= %d \n", (*fp)(33,11) );
}


int add(int a, int b) { return a+b; }
int sub(int a, int b) { return a-b; }
int mul(int a, int b) { return a*b; }
int div(int a, int b) { return a/b; }
```

f4.c

```
#include <stdio.h>

int psum(int n) {
  int k, S=0;
  for (k=1; k<=n; ++k) S+=k;
  return S;
}

int g = 100;

int main(void) {
  int S1;
  int (*fp) (int n) ;

  fp = psum;


  S1 = fp(g);
  printf("S1=%d\n", S1);

  printf("sizeof(fp)= %ld bytes \n", sizeof(fp));
  printf("fp= %p &fp= %p \n", fp, &fp);
  printf("psum= %p &psum= %p \n", psum, &psum);
  printf("main= %p \n", main);

  return 0;
}
```